Coarse Grain Reconfigurable ASIC through Multiplexer Based Switches

Karen Gettings, Marc Burke, Jeremy Muldavin, Michael Vai
MIT Licoln Laboratory
Lexington, Massachusetts, USA
{karen.gettings, marc.burke, muldavin, mvai}@ll.mit.edu

Distribution A: Public Release

***Abstract*—We present an ASIC architecture with coarse grain reconfigurability, by using accelerators to improve performance over fine grain reconfigurable architectures. A reconfigurable FFT ASIC was built as a proof of concept, and it successfully proved the switch implementation.**

***Keywords*—Low power, low SWaP, reconfigurable, coarse grain, FFT.**

I. Introduction

ASIC (application specific integrated circuit) implementations are particularly attractive for applications with tight size, weight and power constraints. ASIC technology has a 10-1000X performance advantage over FPGAs and GPPs [1-2], but designers shy away from ASICs as they are expensive, inflexible, and slow to fabricate.

Figure 1 summarizes the performance and flexibility of three embedded processing techniques: FPGAs (field programmable gate arrays), GPPs (general programmable processors) and ASIC (application specific integrated circuit). Unfortunately, performance and flexibility are mutually exclusive, as indicated in Figure 1 by the data points collected from representative applications. A one-size-fits-all, high-performance embedded processor is unachievable. Our goal is to develop domain-specific embedded processors that have ASIC-like performance and FPGA-like flexibility.

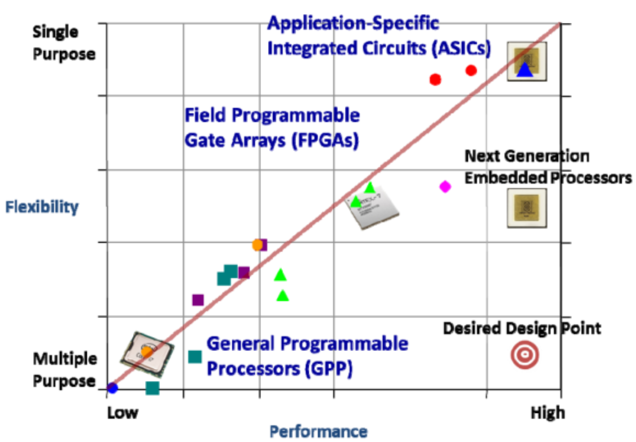

Figure 1: Embedded processing system design space.

Furthermore, throughout history the clock speed of general processors improve drastically year after year, which discourage some to spend the resources of developing ASICs because given the long design and implementation cycles for ASICs, it is likely that the COTS processor would catch up to the performance of the ASIC part. However, since 2000, the clock rates of COTS processors have not been increasing and this trend is expected to continue, thus preserving the advantages of ASIC over COTS. This is illustrated in Figure 2.

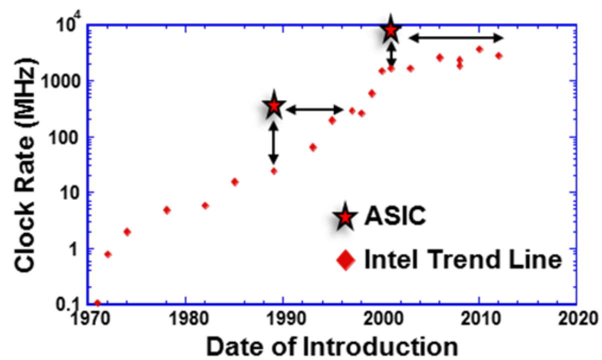

Figure 2: Trend of clock rates in general purpose processors and ASICs.

One way to reduce the time and cost of a system but still have the performance of ASIC accelerators is by creating a reconfigurable ASIC. A reconfigurable ASIC allows ASIC like performance by implementing highly optimized kernels that can be accessed and configured via switches. The kernels can be connected to implement a specific set of functions. Coarse grain reconfiguration assures that the kernels are optimized for a particular function and that the configuration on the chip can be changed quickly, at the expense of the added flexibility of fine grain reconfigurable structures like FPGA.

As a proof of concept of what a reconfigurable ASIC would look like we built a reconfigurable Fast Fourier Transform (FFT) ASIC chip capable of performing FFTs of various sizes. We chose an FFT chip because FFTs are ubiquitous on communication, radar and other application domains. It is also an area where we can gain a lot of power performance by using ASICs. Figure 3 illustrates the increased performance of a small FFT block in an ASIC architecture vs. other platforms. It also shows the drawbacks of decreased flexibility for the higher performance platforms.

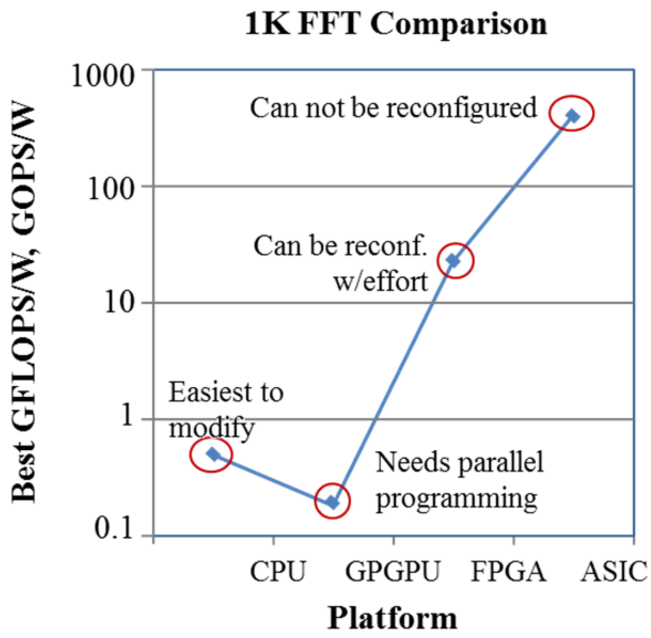

Figure 3: Comparison of 1K FFT implementation on various platforms.

II. Reconfigurable FFT ASIC Design

A. Switch Network

The coarse grain configuration is facilitated by low power, low area switches. For the test chip we decided to implement unidirectional multiplexer-based switches vs. tri-state buffers. Based on the research by [3], unidirectional switches have the following benefits:

- Simplified circuitry for drivers – no tristate buffers required
- Reduced capacitance on routing wires due to shorter wires and smaller loads
- Net improvement in area-delay product.

The analysis performed by [3] also shows that the area and power penalty for using multiplexers vs. tri-state buffers is negligible when taking into account the buffering needed to recover the signal from the losses in tri-states. Furthermore, because we are trying to build a chip that is easily expandable in the future, the tri-states would have imposed limitations on the distance between the switches. The multiplexers don't have an issue with signal levels or need recovery, and they pose no risk in timing closure with the automated synthesis and place and route.

By using coarse grain reconfiguration we avoid common routing issues that are present with fine grain reconfigurable structures, like FPGA. Figure 4 shows a block diagram of the switches. By limiting the number of connections and programming the switches before operation start we can achieve low level of congestion between the accelerators. There are two paths for the switches: nearest neighbor or "long distance." The limited paths help with timing closure of the implementation.

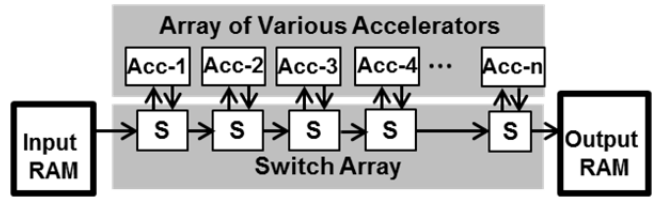

Figure 4: Block diagram of reconfigurable architecture with accelerators.

B. FFT Kernel Building Block

We implemented various FFT modules into an FFT kernel array to accommodate of the desired various FFT sizes, partially based on the work by [4] and [5].

The dense FFT kernel block is illustrated in Figure 5. The FFT blocks includes a radix-2^2 single delay RAM feedback stages. The RAM delay structure minimizes the number of interconnects required between stages, reducing the size and complexity of the switch network. The RAM structure allows per-stage configuration, providing the use a single common FFT block for the array.

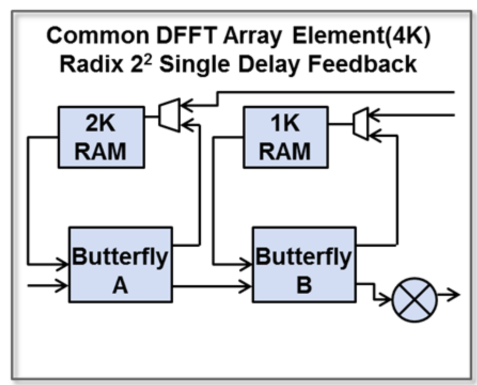

Figure 5: Common Dense FFT array element

Configuration options to support the various dense FFT sizes are shown in Figure 6.

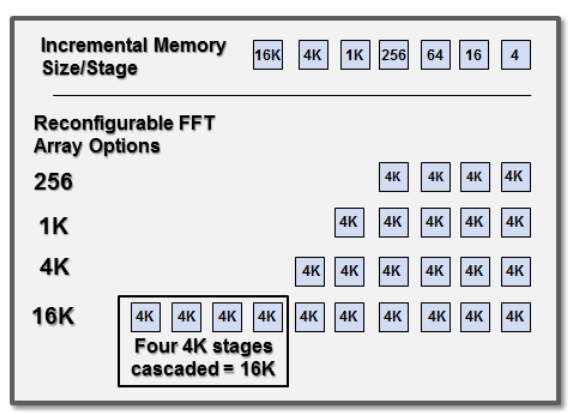

Figure 6: Dense FFT configuration options

This work is sponsored by the Department of the Air Force under Air Force Contract FA8721-05-C-0002. Opinions, interpretations, conclusions and recommendations are those of the authors and are not necessarily endorsed by the United States Government.

The incremental memory size per stage for a Radix-2^2 single delay feedback 16K FFT structure is located at the top of Figure 6. For example, to support a 16K FFT, seven stages are required, each sized from 16K down to 4. Our array of FFT modules, sized at 4K each, supports up to 4K FFTs naturally. To support FFTs larger than the natural size (16K for example), support has been provided to cascade the memory portions of the FFT blocks, not utilizing the logic on these blocks. For 16K support, four FFT blocks are interconnected, three of which are configured to supply memory function only, as shown in Figure 6.

Area efficiency vs. FFT RAM size has been studied for a range of FFT sizes. Figure 7 illustrates the tradeoff in area used when selecting the memory size of the common FFT array module. The blue line illustrates the incrementally sized FFT structure (optimal). For FFT sizes from 1K to 64K, our analysis concluded that a common 4K stage was optimal among the reconfigurable options. To further reduce memory requirements for the common module, the twiddle factor RAM was reduced by a factor of 8 by exploiting the symmetry in twiddle factor generation. Figure 7 also illustrates the penalty in area for reconfiguration due to the accelerators. To make the accelerator flexible we needed a larger area than we would have otherwise needed for a non-reconfigurable kernel. Larger area also results in a more complex clock distribution and hence slightly increased power consumption, despite the use of clock gating. While the penalty for both area and power is negligible for the switches, careful implementation of flexible accelerators is imperative to maintain high performance in a reconfigurable chip.

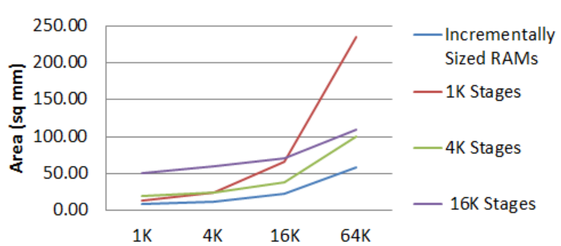

Figure 7: 180 nm CMOS implementation of FFT stage size vs. area.

III. Reconfigurable FFT ASIC Implementation and Results

We implemented and taped out a reconfigurable FFT ASIC test chip in 180 nm CMOS technology. The chip supports 256, 1K, 4K, and 16K FFT sizes, as well as an exploratory sparse FFT implementation based on the work by [5]. Figure 8 illustrates the physical implementation.

The 180 nm process has 6 metal layers, and all were used in the design. The size of the chip is 9.5 mm by 10.5 mm. The regular FFT blocks (DFFT) occupy approximately half of the chip area, while the remaining sparse FFT modules and IO buffers occupy most of the other half.

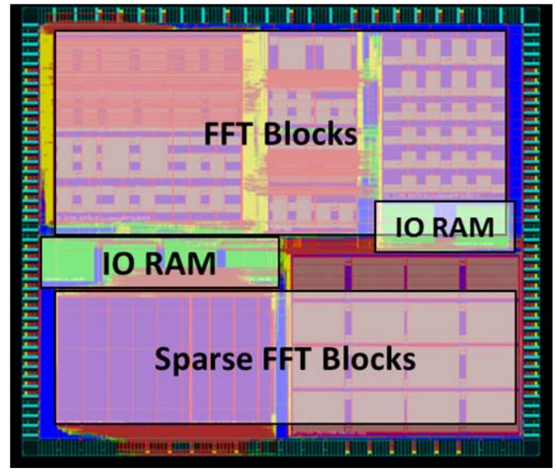

Figure 8: CAD plot of 180 nm reconfigurable FFT ASIC.

The switches use less than 1% of the chip area (0.5 mm^2), and from simulation their power consumption is negligible (0.002% from simulation, too small to measure in physical system).

Switches can be programmed at the full speed for which they were designed in this proof of concept, 50 MHz, and up to 50% faster speeds were also tested successfully.

The performance of the chip was also characterized as taking various paths, as illustrated on Figure 9. Path 1 is a nearest neighbor path, and Path 2 is a serpentine between the "long distance" switch array and the nearest neighbor switch array. Our chip could successfully navigate both paths at full speed (50 MHz), with no impact to functionality.

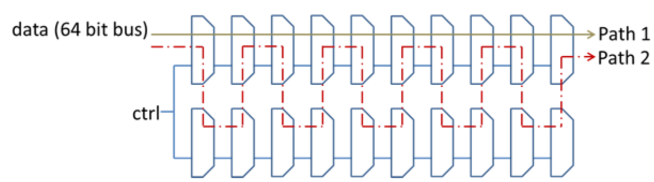

Figure 9: Tested paths on switch network.

IV. Future Work

We believe that the switch functionality can be enhanced by adding optional registers between a set number of multiplexers. This will enable easier timing closure at the expense of latency. This tradeoff must be assessed for a particular set of applications.

V. Summary

We have successfully implemented a switch array to enable coarse grain reconfiguration into an ASIC with highly specialized accelerators. Using multiplexers as the building block allow for a more efficient implementation than traditionally reprogrammable platforms, like CPUs and FPGAs, but more flexibility than using a traditional ASIC. By using multiplexer based switches and minimizing the routing we minimally impacted the area and performance of the chip, enabling a high performing implementation that is also flexible.

REFERENCES

[1] I. Kuon and J. Rose, “Measuring the gap between FPGAs and ASICs,” IEEE Trans. Computer-Aided Design, vol. 26, no. 2, pp. 203–215, Feb 2007.

[2] A. Reuther, "Power Analysis of Key Computational Kernels (PAKCK)," Suite of Embedded Applications and Kernels (SEAK) Workshop at the Design Automation Conference (DAC), San Francisco, CA, June 1, 2014.

[3] P. Grossman, M. Lesser, “Circuit-Level Considerations for an Ultra-Low Voltage FPGA with Unidirectional, Single-Driver Routing Fabric,” Presentation at IEEE Subthreshold Microelectronics Conference, 2011.

[4] Chia-Hsiang Yang; Tsung-Han Yu; Markovic, D., "Power and Area Minimization of Reconfigurable FFT Processors: A 3GPP-LTE Example," *Solid-State Circuits, IEEE Journal of* , vol.47, no.3, pp.757,768, March 2012.

[5] Agarwal, A.; Hassanieh, H.; Abari, O.; Hamed, E.; Katabi, D.; Arvind, "High-throughput implementation of a million-point sparse Fourier Transform," *Field Programmable Logic and Applications (FPL), 2014 24th International Conference on* , vol., no., pp.1,6, 2-4 Sept. 2014.